Domande/Risposte per TecWeb

Http1.0 vs Http1.1 vs Http1.1 con pipeling ?
La prima manda una sola req/res per connessione e quindi non è una connessione persistente.
In entrambe possiamo fare più req/res in una sola connessione e quindi parliamo di connessioni persistenti. Nel primo caso però possono essere effettuare solo req/res in questo ordine, mentre nel secondo caso possiamo inviare molteplici req di seguito e poi ottenere le rispettive risposte nello stesso ordine.

Flag 'secure' nei cookie ?
Nei cookie, il flag 'secure' serve per indicare se vogliamo che il cookie venga inviato anche se la comunicazione non è sicura.

Session bean (stateless vs stateful) ?
I SessionBean non sono persistenti e non rappresentano i dati in un DB ma un processo di business. Per ogni cliente abbiamo una loro nuova istanza e muoiono col cliente.
Quelli stateless eseguono una richiesta/risposta senza salvare informazioni del cliente, al contrario di quelli stateful.

Activation vs Passivation ?
Vuol dire che stiamo parlando di stateful session bean, e nello specifico dei 2 modi per gestire la coppia (EJB, istanza bean stateful).
Con la Passivation abbiamo la dissociazione della coppia e il salvataggio dell'istanza in memoria.
Con la Activation abbiamo la riassociazione della coppia, prendendo dalla memoria l'istanza.

Libreria asincrona in node.js ?
async

Include vs Forward nelle servlet ?
L'inclusione di una risorsa indica che questa può lavorare con il response-body da quando viene inclusa a quando rilascia l'output che viene poi ripreso dal server che può continuare la sua esecuzione.
L'inoltro di una risorsa indica invece che questa è l'unica responsabile del response-body che riceverà il cliente (che non si accorgerà del cambio di server), l'output viene quindi definitivamente rilasciato.
Nelle servlet, esiste l'oggetto *RequestDispatcher* che offre i metodi *include(req, res)* e *forward(req, res)*.

Pro/Contro di JSP e Servlet ? {MIGLIORABILE}
1. JSP
   - 🔴 Difficile da testare
   - 🔴 Non si ha il completo controllo dell'applicazione

- 🟢 Divide bene lavoro tra grafici e programmatori
- 🟢 Rende facile rappresentare documenti HTML e XML

2. Servlet
   - 🔴 Generazione pagine resp lento e imperfetto
   - 🔴 Aggiornamento scomodo delle pagine
   - 🟢 Forniscono completo controllo dell'applicazione
   - 🟢 Facile da testare

Da cosa deriva il formato JSON ?
Il formato JSON deriva dalla necessità di trovare un metodo di formattazione del contenuto più ottimizzato di XML e per fare ciò si basa sulla notazione per le costati oggetto e array in JavaScript.

Cos'è JM1 e da cosa è composto ?
È un modello di sviluppo per applicazioni in java, prevede servlet e jsp. Utile per applicazioni piccole. La logica di business e di presentazione non sono separate. Sta divento obsoleto.
Le servlet fungono da controller e interagiscono col modello (JavaBean).
Le JSP fungono da view per l'interfaccia utente.

Cos'è CGI ?
CGI è uno standard per interfacciare applicazioni esterne con Web Server e rappresenta la prima soluzione proposta per ottenere il Web Dinamico in quanto è un programma eseguito in risposta alla chiamata e produce un output che costituisce la risposta.
Il server e CGI comunicano tramite variabili d'ambiente, parametri di command-line, stdIn e stdOut.

Cos'è un ipertesto?
Un ipertesto è un insieme di documenti messi in relazione tramite collegamenti monodirezionali (link) che permettono una lettura non lineare.
Gli elementi di un ipertesto sono l'URI (lo identifica), HTTP (permette comuicazione) e l'HTML (ne descrive il contenuto).
L'ipertesto è l'oggetto che viene scambiato in una comune interazione C/S.

Cos'è un Application Server ?
Un application server è un contenitore con tutte le funzioni server-side. Gestisce l'interfacciamento con il WebServer, il tempo di vita, l'interfacciamento col database e la sicurezza.

Come viene gestita una richiesta da un server?
Il client invia una richiesta. Questa viene prima ricevuta dal container, il quale si occupa di mantenere lo stato, del sistema di nomi e di garantire la QoS. La richiesta a questo punto può essere trattata dal server applicativo il quale la legge, elabora una risposta con una sua logica e restituisce la risposta.

Pro/Contro di AJAX ?

🟢 Sulla stessa pagina ci possono essere numerose richieste indipendenti
🟢 Non si avverte il tempo di attesa
🔴 Si potrebbe percepire che non stia accadendo nulla se non comunichiamo nulla
🔴 Difficile gestire elaborazione sincrona

Cosa significa 'cascade' in css ?
Come dice anche il nome, Cascading Style Sheets, in css i fogli di stile hanno un effetto a cascata, ovvero possono essere applicati uno dopo l'altro in modo indipendente.
Per garantire l'indipendenza, è stata creato uno standard 'cascade' che si occupa della risoluzione dei conflitti di stile. Per fare ciò assegna a ciascun blocco di regole un punteggio che si calcola in base all'ORIGINE (Autore, Browser, Utente), all'SPECIFICITÀ (id > class), ORDINE (vince l'ultima) e alla presenza o meno della clasusola *!important*. Il blocco col punteggio maggiore vince il conflitto.

Cos'è l'Applicazione 3-tier ?
È un'architettura frequente nei sistemi Web che rispecchia i 3 principali servizi che vengono richiesti a un sistema Web.
Può avere un'architettura verticale, dove i 3 servizi risiedono sulla stessa macchina (fault tolerance non-ok), oppure un'archiettura orizzontale, dove ogni servizio è replicato su diverse macchine (fault tolerance ok).

Quali sono i bound ?
??

Come funziona il Virtual-DOM di React ?
Il Virtual-DOM è una copia in memoria centrale del DOM di un browser, egli permette una manipolazione più leggera in quanto solo alla fine delle operazioni sul VDOM vengono individuate le differenze rispetto al DOM e aggiornate. Il rendering della pagina risulta così più leggero, efficiente e veloce.

Come fa React a capire quali elementi eliminare ?
Ad ogni aggiornamento dello stato di un componente viene creato il VDOM, questo viene modificato dalle varie modifiche dello stato. Una volta finite le modifiche, viene confrontato con il DOM e viene creato un elenco di modifiche minime da apportare poi al DOM in base alle differenze (diffing).

Cos'è HTTP ?
HTTP è il protocollo applicativo per trasferire risorse Web ed è quello che trasporta richieste e risposte nel modello C/S.
Si basa su TCP per garantire la consegna ordinate e completa dei messaggi, vogliamo la QoS (UPD non è affidabile).
I messaggi che trasporta sono composti da Message header (insieme di coppie-valore su trasmissione, entità trasmessa, richiesta. risposta) e da Message Body (dati veri e propri).

Cos'è HTTPS ?
HTTPS è il protocollo HTTP che però lavora con un canale TLS più sicuro, esso è un livello in più che si occupa della confidenzialità, autenticità e integrità della comunicazione tra HTTP e TCP tramite certificati e chiavi pubbliche/private.

All'interno di una Servlet posso vedere un Bean con scope session definito in una JSP? E come posso accedervi ?
Se in una JSP ho un Bean con session-scope e lo stesso utente della JSP fa una richiesta alla Servlet allora condivideranno la stessa sessione e quindi tramite l'oggetto *(MyBean) HttpSession.getAttribute("MyBean")* nella Servlet potrò accedere al Bean.

Scope (page vs request vs application vs session) ?
I JavaBean con scope:

1. page: accessibile solo nella pagina corrente e dura fino al completamento della pagina.
2. request: accessibile alla pagina corrente + incluse/delegate e dura fino a restituzione della risposta
3. session: accessibile a tutte le richieste dello stesso client e muore con lui
4. application: accessibile finchè l'applicazione è attiva e muore con lei

Come faccio una Web Application visualizzabile su dispositivi diversi ?
Usando diversi file css.

In React, come si strutturano le applicazioni ? DA FINIRE
Si devono includere i componenti .js in un html dove viene posizionato un componente principale App.js che al suo interno farà il render degli altri componenti ricorsivamente.

In React, come triggero (attivo) la modifica della visualizzazione?
Si deve modificare lo stato del componente logico tramite setState(). Così facendo viene modificata la logica dell'applicazione, che quindi fornirà proprietà diverse ai sottocomponenti visivi che quindi modificheranno la visualizzazione.

Cosa sono le operazioni idempotenti ?
Sono operazioni stateless, ovvero che allo stesso input rispondono con lo stesso output.

Cos'è Node.js in generale ?
??

HTML vs SGML ?
Sono entrambi linguaggi dichiarativi. Solo che HTML è una semplificazione di SGML in quanto entrambi definiscono vari elementi definiti da tag iniziali e finali, ma SGML ha una sintassi più generica per definire in generale documenti

elettronici mentre HTML ha il solo scopo di rappresentare pagine web e quindi la sua sintassi è più specifica. Inoltre HTML introduce la rappresentazione grafica dei suoi elementi, non presente in SGML.

Come sono composti i campi di request/response ?

1. Response
   - status line —> protocollo + status code + status phrase
   - header —> connection + server + content_lentgth + content_type
   - response body
2. Request
   - status line —> tipo_richietsa + URL + protocollo
   - header —> host_name + authorization + cookie + user_agent
   - response body —> vuoto se GET

In WebSocket, come fa il server a iniziare una connessione ? DA CONTROLLARE

Il server di default non conosce l'indirizzo del client in quanto la struttura C/S è asimmetrica

Cosa sono gli URI? Codice fiscale e latitudine/longitudine sono URL o URN ?

Gli URI sono Uniform Resoiurces Identifier, ovvero oggetti che servono a identificare una risorsa web e si dividono in URL (identificano la risorsa per la sua locazione) e URN (identificano la risorsa per il suo nome).
Quindi URL=latitudine/longitudine e URN=codice_fiscale

In AJAX, come vengono gestite le richieste sincrone e asincrone ?

Le richieste vengono modellate dall'oggetto nativo *XMLHttpRequest* e vengono iniziate da
open(metodo, uri, true/false) dove true=asincrono e false=sincrono
A questo punto tramite *setRequestHeader(nomeheader, valore)* setto il request_header
Infine invio la richiesta con *send(body)*

In CSS, quali caratteristiche un figlio eredita dal padre ?

Per l'ereditarietà, uno stile applicato ad un blocco di applica ai blocchi in esso contenuti e quindi un figlio eredita gli stili del padre tranne gli stili di formattazione del box-model, per evitare una grafica a matrioska

Pro/Contro di singlethread e multithread ?

Se vengono fatte molte richieste dello stesso tipo al server e questo è singlethread allora avremo la creazione delle stesse strutture più volte (🔴) mentre se questo è multithread avremo che crea un thread per ogni richiesta e le strutture saranno condivise tra i thread (🟢) anche se andrà poi gestita la concorrenza che ne deriva (🔴).

Pro/Contro di container leggero e pesante ?

Sono strumenti di virtualizzazione/isolamento per l'esecuzione di applicazioni.

1. Container leggero -> istanza di applicazione virtualizzata, condivide kernel
   - 🟢 Efficienza delle risorse, meno overhead perchè meno risorse duplicate
   - 🟢 Avvio rapido, bisogna istanziare meno risorse all'avvio
   - 🟢 Scalabilità, essendo leggeri posso usarne tanti
   - 🟢 + container x host, deriva da scalabilità
   - 🔴 Limitazioni kernel, bisogna tenere quello dell'host
   - 🔴 Isolamento limitato, si dipende da kernel
2. Container pesante -> istanza di applicazione virtualizzata, ha kernel separato
   - 🟢 Isolamento completo, non dipendono da kernel host
   - 🟢 Flessibilità utilizzo kernel, possono usare kernel come vogliono
   - 🟢 Facilità di migrazione, sono autocontenuti
   - 🔴 Maggiori risorse richieste, servono più risorse da istanziare per il funzionamento
   - 🔴 Avvio lento, all'avvio sono richieste più risorse
   - 🔴 Scalabilità limitata, essendo pensante il carico diventa gravoso su larga scala

GET vs POST ? {MIGLIORABILE}

1. GET
   - parametri della richiesta nell'header —> dimensione limitata (255 caratteri)
   - utile per chiedere una risorsa
2. POST
   - parametri della richiesta nel body —> nessun limite
   - utile per sottomettere i dati privati

Quali sono i vantaggi dei fogli di stile piuttosto che codice inline ?
Il vantaggio dei fogli di stile CSS rispetto al definire lo stile direttamente inline nell'HTML sono che il codice diventa più mantenibile, abbiamo una più efficiente divisione del lavoro tra programmatori e grafici e così non dobbiamo fare una pagina HTML diversa per ogni pagina con stile diverso.

Quali sono i meccanismi di mantenimento dello stato? Pro/Contro ?
I due metodi sono cookie e sessione server-side.

1. Cookie
   - 🔴 Sicurezza, sono vulnerabili ad attacchi di furto e falsificazione
   - 🔴 Persistenza limitata, capacità di archiviazione dipende dal browser
   - 🔴 Dipendenza dal browser, se in un browser i cookie sono disabilitati viene compromessa l'applicazione.
   - 🟢 Persistenza, sono memorizzati nel browser e quindi sopravvivono alla sua chiusura.
   - 🟢 Scalabilità, non richiedono risorse server-side per il mantenimento
   - 🟢 Personalizzazione, permettono di personalizzare l'esperienza dell'utente

2. Sessione
    - 🟢 Sicurezza, i dati sono memorizzati server-side
    - 🟢 Persistenza illimitata, non vi è un vero limite all'archiviazione
    - 🟢 Indipendenza dal browser del client
    - 🔴 Scalabilità, per molti clienti il peso sul server diventa importante
    - 🔴 Complessità, la logica lato server dovrà gestire le associazioni sessione-richiesta
    - 🔴 Durata, sessione scade dopo periodo di inattività

Cos'è una sessione in una Web Application ?
La sessione è uno dei modi di mantenere lo stato dell'applicazione, memorizzando in appositi oggetti i dati di tutte le interazioni con uno specifico utente.
Lo scope è dato dal tempo di vita (durata dell'interazione utente) e dall'accessibilità.
È il concetto alla base di conversazione, ovvero la sequenza di pagine e di interfacce di comunicazione.
insieme dei dati che caratterizzano un'interazione con uno specifico utente.

Cos'è uno scope-session in un JavaBean ?
Lo scope-session in un JavaBean indica che quel Bean sarà accessibile solo nella stessa sessione.

Quali sono le tecniche di aggiornamento della cache ?
Per aggiornare la cache possiamo usare la Freshness (ovvero settare sia lato client/server delle durate ai documenti in cache), la Validation (tramite richieste HEAD, verifichiamo se documenti sono validi) e Invalidation (se per la cache passa una nuova versione del documento allora questo viene salvato e l'altro invalidato).

Cos'è JM2? Come viene applicato nell'architettura EJB ?
Java Model 2 è un modello di sviluppo di applicazioni java. La logica di business e di presentazione sono separate. Utile per applicazioni di grandi dimensioni. Molto simile a MVC.
EJB offre supporto per la logica di business e l'accesso ai dati.
C'è anche JSF che offre supporto per la creazione di interfaccia e interazioni con utente.

Proxy vs ReverseProxy ?
Un Proxy è un programma applicativo che compie richieste per conto di altri clienti, svolgendo sia il ruolo di serve che di client.
Un Reverse Proxy è un programma applicativo che funge da intermediario per altri server, funziona al contrario del Proxy, nascondendo quindi il Server e non il Client.

Proxy_cache vs User_cache ?
Le Proxy Cache si dividono in Forward (intercetta le pagine e le mette in cache)

e Reverse (pagine chieste al server vengono cachate per richieste successive che non impegneranno il server).
Lo User Agent Cache mantiene in memoria le pagine del browser, consentendo di ottenere pagine anche senza connessione.

Cos'è un Iframe ?
Sta per Inline Frame, ovvero un altro frame nella stessa pagine, dentro il quale può essere visualizzato un altro documento HTML.

Cosa succede quando si clicca su un ancora, quindi un link ?
Se l'ancora di partenza puntava a un'ancora di ritorno locale ci sposteremo verso di esso, altrimenti se l'ancora di ritorno si trova in un altro documento allora viene fatta una GET.

Cos'è il context?
Rappresenta il contesto di esecuzione della Web Application, dove possiamo salvare degli attributi visibili a tutte le servlet del contesto e che fungono da variabili globali.

Web socket definizione, streaming forever con e senza le socket
WebSocket è un protocollo di comunicazione bidirezionale e full-duplex che consente una comunicazione interattiva in tempo reale tra client e server tramite una connessione persistente e stateful.
Streming forever -> client si iscrive ai messaggi del server

- con WS, si usa stream normalmente
- senza WS, posso fare richieste HTTP frequenti

Spiega la libreria fs e callback node
...

Cos'è la Dependency Injection in Ejb 3.0 ?
La Dependency Injection è una tecnica per gestire/risolvere dipendenze tra componenti EJB, rendendo l'applicazione più flessibile (no lookup in business), testabile (testing automatico, no dipendenze) e mantenibile (ambienti applicativi sono riutilizzabili solo cambiano conf).
La logica è quella di creare oggetti/componenti quando necessario e di passarli automaticamente agli oggetti/componenti che li devono utilizzare.
Può essere implementata a livello di costruttore (costruttore non vuoto) o a livello di setter (come spesso nei bean).

Cosa si usa per configurare in ejb 3.0 al posto di xml ?
Invece di usare dei file di configurazione in xml possiamo applicare delle annotazioni come @Stateless o @Stateful per definire le proprietà del Bean, rendendo la configurazione più semplice e leggibile e velocizzando lo sviluppo.

Come faccio ad aumentare le performance di applicazione con Web server, application server e db server ?

Per il WebServer, posso utilizzare caching per evitare richieste inutili opppure posso distribuire orizzontalmente per avere un load-balaning ottimale.
Per l'Application Server, usare il cache per le operazioni più costose oppure usare una programmazione asincrona per gestire richieste concorrenti.
Per il Database Server, potrei minimizzare lo scambio di traffico e usare query più efficienti.

Cosa succede ai dati di sessione se l'application server va giù e come faccio ad evitare di perdere i dati ?
I dati vengono persi e per evitare la perdita bisogna valutare di salvare i dati in un database o in un server esterno.

Differenze fra thread singolo di Node e multi-thread di Java Servlet (problemi del multi-thread) ?
...

In React, cosa sono props,states e render ?
Le props sono le proprietà che vengono passate a un componente alla sua creazione, sono immutabili.
Lo state è lo stato che viene dato a un componente, è come props ma modificabile (solo da sè stesso) tramite setState() e inizializzabile nel costruttore.
setState() provocherà nel componente l'invocazione di render().

Quali sono i componenti di ejb 3.0 e a cosa corrispondono tra servlet jsp e java bean ?
I componenti di EJB 3.0 sono SessionBean + EntityBean (comunicazione sincrona) e Message-DrivenBean (comunicazione asincrona).
Corrispondono alle Servlet.

Nelle WebSocket come viene effettuato l'upgrade ?
L'upgrade è il processo che trasforma una tradizionale comunicazione HTTP in una connessione WebSocket, consentendo una connessione bidirezionale e persistente (messaggi real-time e asincroni).
Protocollo di handshake:

1. il client richiede tramite header HTTP settato [Upgrade="websocket", Connetion="upgrade"].
2. Se il server lo supporta, risponde con una conferma HTTP e qui abbiamo la connessione stabilita.
3. WebSocket è funzionante e attiva finchè non viene chiusa o non ci sono errori.

In AJAX, cosa succede dopo la send() ?
Dopo la send(), l'oggetto XMLHttpRequest invia la richiesta al server e non si blocca in attesa dei risultati. Infatti prima di fare la send() bisogna impostare una funzione per la gestione della risposta in base al cambiamento dello stato di XMLHttpRequest, e quando lo stato sarà 200 allora potremmo trattare la

risposta nel responseText.

In Node, cosa lo rende diverso dagli altri web server ?
...

Un componente spring può usare un altro componente spring ?
si, tramite <ref>

Qual è il ciclo di vita di una JSP ?
Il ciclo di vita e:

1. Parsing : alla richiesta, la jsp viene analizzata dal container -> diventa servlet
2. Compilazione : viene compilato il codice java
3. Inizializzazione : la prima volta [init()]
4. Esecuzione : viene ricevuta req e prodotta res [service()]
5. Distruzione : viene distrutta quando inutile [destroy()]
6. Rilascio : dopo essere stata distrutta, ne vengono rilasciate le risorse tramite il GarbageCollector

In WebSocket, un Client può mandare in broadcast un msg ?
Non direttamente, il client è connesso solo al server e quindi deve passare il messaggio prima a lui indicandogli di mandarlo in broadcast.

Polling vs Long Polling ?
Nel polling il client effettua periodicamente richieste al server per ottenere i dati aggiornati, risulta poco efficace e gravoso se la frequenza è elevata (molto spesso server non ha la risposta).
Nel long-polling il client invia un messaggio e il server mantiene aperta la connessione finchè non ci sono i dati di interesse del client.

ManagedBean in JSF ?
JSF è un'evoluzione a componenti di JSP.
Il componente, ManagedBean, tramite l'annotazione @ManagedBean registra automaticamente il componente come risorsa utilizzabile all'interno del container JSF.
In sostanza sono semplici JavaBean che seguono regole standard:
- Costruttore senza argomenti (empty)
- No variabili di istanza public
- Metodi "accessor" per evitare accesso diretto
- Getter/Setter
La novità sta nella possibilità di avere anche il bean come scopeless

Ciclo di vita di una JSF ?

1. Restore View : prende la view precedentemente associata all'utente che fa richiesta
2. Apply Request : ottiene i valori della richieste li associa alle proprietà dei ManagedBean

3. Process Validation : verifica i dati inseriti e memorizza eventuali errori
4. Update Model Values : se i valori sono validi allora vengono effettivamente inseriti nel ManagedBean
5. Invoke Application : vengono eseguite logiche di business, tramite i ManagedBean
6. Render Response : viene generata risposta HTML sulla base dello stato gestito dai ManagedBean